Which Fortify tool should I use to scan my application

This page has been made public for vendors

Question

Fortify provides several tools to scan an application. Which one should I use for my application?

Answer

Fortify provides a variety of command-line, GUI, and build environment tools to scan an application. For most applications there are multiple ways to scan an application. All the scan methods use the `sourceanalyzer` tool so given the same inputs they will all produce the same output.

There are several criteria you should consider when choosing a scanning tool:

- Language(s) in the application to be scanned
- Regular build environment
- Automation requirements
- Scan repeatability

This remainder of this technical note will look at selecting a tool based on these criteria and a look at strengths and weaknesses of each tool.

Tools

The following tools are available to scan an application:

- **Command-line tools** - The `sourceanalyzer` command-line tool can be used to scan any codebase as all the other tools are based on this tool. It is a good choice to use to build scanning scripts that can be reused for consistent scans or integrated into development environments to automate scanning. The downside is that for some applications it can be difficult to set the command lines necessary to successfully scan the applications.
- **Scan Wizard** - The Scan Wizard is a GUI tool that provides a step-by-step guide to creating a scanning script (either a batch file or shell script). It facilitates use of the command-line tools and therefore has many of the advantages and helps reduce the difficulty in using `sourceanalyzer`. The Scan Wizard cannot be used to create scanning scripts for compiled languages which it doesn't have a built-in compiler (e.g., C/C++, Objective-C).
- **IDE Plugins** - Fortify comes with plugins for Visual Studio and Eclipse. With the plugins, Fortify scans can be run from a menu item and it will use information from the Visual Studio solution or Eclipse project to help ensure a complete scan is performed.
- **Build Environment Integration** - Fortify provides tools to integrate scans into many build environments. Fortify provides a plugin to integrate with Maven and an Ant task to integrate with Ant. Fortify can be integrated either directly with MSBuild, Makefile, and other build environments or the "touchless build adapter" can be used to utilize these build environments without modifying the build files.
- **Audit Workbench** - The primary purpose for this tool is to display and audit FPR files. It can also be used to perform scans through a series of questions to establish the codebase and how to scan it. This is similar to the Scan Wizard except the tool scans the application instead of production a scanning script. It also has similar limitations as to which languages it can scan.

More details about the tools and how to use them are included in the documentation provided with the Fortify.

Selecting a Tool

The first consideration when selecting a scanning method is how do you normally compile your code? If you normally compile using Visual Studio or Eclipse, the plugins for those tools are likely the best tool to use to scan the code. Similarly if you use Ant or Maven the Ant task or Maven plugin will likely provide the easiest path to scan the code.

For other compiled languages (except for Java), scan options are limited. They must be scanned either by integrating `sourceanalyzer` into the build files or use the "touchless

HPE Fortify Version	16.11 and later
Programming Language	☑ C/C++ ☑ .NET ☑ Java ☑ Objective-C ☑ Other
Fortify Audit Workbench	☑ Yes ☐ No
Fortify IDE Plugin	☑ Yes ☐ No
Other Fortify Component	☑ Yes ☐ No

Request code review tools, validations, and support HERE.

build adapter" option to `sourceanalyzer` while scanning on the command-line.

Most languages supported by Fortify, except those like C/C++ and Objective-C that depend on outside compilers, can either be scanned by creating a scanning script with the Scan Wizard or directly in Audit Workbench. Of these two, using the Scan Wizard is likely the better option as it produces a script that can be reused for scanning. In Audit Workbench the scanning options must be provided for each scan which can lead to failing to perform the scan the same way each time it is run. This is especially true if you have many customizations beyond a default scan.

If you want to integrate Fortify into a larger automated build environment, it is likely either working with the command-line tools directly or starting with a scan script produced by the Scan Wizard will be necessary to integrate appropriately. Note for this kind of automation, Fortify provides additional command-line tools to merge FPRs and perform other tasks available in the GUI interfaces.

References

- HPE Fortify Static Code Analyzer User Guide